Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 1 (2012) 474 – 480

Procedia Technology

INSODE 2011

Comparing text classifiers for sports news

Tarik S. Zakzouk [a*] and Hassan I. Mathkour [b]

[a] *Ph.D Student, Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia*
[b] *Professor, Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia*

Abstract

The rise of Machine Learning (ML) techniques has given life to text classification discipline and its many applications we enjoy these days such as spam filters and opinion mining which became an essential part of our daily life. Tools and techniques have shown tremendous maturity especially in the past two decades. In this paper, we revisit this field using both commodity software and hardware to show progress of both efficiency and effectiveness of a group of ML-based methods in classifying Cricket sports news articles.

Keywords: NLP; Text Classification; SVM; C4.5; Naïve-Bayesian;

1. Introduction

The Area of text categorization has been a hot topic for the past 20 years. Although methods and techniques have matured through that time but the needs are still growing and applications are never ending. Many research studies have been produced comparing among various machine learning based classifiers [6, 7]. The classical application of text categorization was mainly against news corpora. SGSC (Saudi Gazette Sports Corpus) is a textual sports news corpus specifically built for research purposes [5]. It has been motivated by Sports, Medicine, and Religion news classification results [1] Compared to famous text corpora such as Reuters [2, 3], it has small size (comprised of only 797 news texts) specialized only in sports. The news text size is found to be between 0.5 KB up to 4.7 KB (3 to 40 lines and between 20 to 900 words not including the title). The following table summarizes the SGSC:

Table 1. SGSC Summary

No. of Documents	Total no. of Words	Without Stop Words	Stemmed Words
797	18,087	17,770	13,632

Within a text news web page, there is very limited meta-data such as the title, author or the news agency. The author (if mentioned) has no fixed location. It is sometimes placed right after the title or at the end of the text. Pictures included are mainly non-relevant to the actual news text. The date is found at the main page URL. The most important missing information is the sports name. The process of building SGSC is described in [5] and has been both manual and slow. A folder is manually created for each day and the news text is mapped (copy and paste) to a

* *Tel.:+966-505-206308 E-mail address*: tzakzouk@gmail.com

doi:10.1016/j.protcy.2012.02.104

single text file having the name as the date-stamp of the news and additional two digits at the end identifying the order of appearance on the website from 1 to 19. After downloading and organizing 2 months of news, all files have to be manually classified and copied into new folders representing the sports they address. This has resulted into having 22 different sports news folders ranging from Cricket (with the highest number of news articles around 178) to Swimming with only one article. This shows that sports coverage of Saudi Gazette is not balanced perhaps due to the nature of the readers segment. More folders were added later to create negative examples for each sport with adequate number of examples. Cricket was chosen to be the sports of choice since it had the majority of the news articles (178 positive examples making 22% of the corpus). Separate binary classifiers were built based on several different machine learning techniques for the purpose of getting the best model.

The remaining of this paper is organized as follows: in section 2, the experiment of building the text classifiers is explained in detail, section 3 describes the result of each classification method, section 4 compares the results of such classifiers, and finally, section 5 gives the summary.

2. The Experiment Setup

The setup for building the classifiers (models) is somehow comprehensive and needs careful design. It involves tool selection, selecting and setting up the right machine learning algorithms, setting up the cases for both learning and testing, and choosing the right measurements for the experiment.

Tool Selection: RapidMiner 5.1 Community Edition with both Text Processing and Weka extensions installed running on UBUNTU 10.04 64 bit Linux was chosen as an example of open source commodity software capable of running on commodity hardware.

Text Classification Methods Selection: Rapidminer provides a variety of them. We have chosen three methods, SVM based on evolutionary algorithm, C4.5, and Naive-Bayesian.

Generating and Applying the Models: Model application was separated from learning to be able to apply to different cases. Models generated from the learning step were applied on fresh data. Results were very promising. In certain cases where feature selection was applied, model application was not straight forward and the new fresh data had to be similarly treated to be able to be handled by the model with some workarounds.

Feature Selection: For each technique, the plain case (no feature selection) is first applied and then three different combinations of feature selection techniques were employed namely:

- Stop Word Removal,
- Stop Word Removal+ Porter Stemming,
- Stop Word Removal + Porter Stemming + Selecting Top 10% Chi-Square Weight Features [4]

A fifth case, aggressive feature selection (Top 1% Chi-Square Feature Weighting) was also tested for some SVM techniques but abandoned due to weak results.

Effectiveness Measures: Four effectiveness measures have been selected which depend on the confusion matrix output, which are: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Those effectiveness measures used in this study are:

- **Precision** (P) = TP / (TP + FP)
- **Recall** (R) = TP / (TP + FN)
- **Accuracy** (A) = (TP + TN) / (TP + TN + FP + FN)
- **F-Measure** (Micro-averaging) = 2 . (P . R) / (P + R)

Efficiency Measures: In addition to effectiveness, the following are efficiency measures are used:

- Total time it takes to produce the model (end to end)
- Size of the produced model (in terms of MB).
- No. of features selected (Dimensions)
- Algorithm specifics such as No. of support vectors for SVM based models.

Corpus Characteristics: Two folders of the SGSC were used containing:

- Positive examples (Cricket documents) with 178 articles
- Negative examples (non-Cricket) with 154 articles.

Both folders have the following characteristics:

Table 2. Corpus Characteristics

No. of Documents	No. of Words	Without Stop Words	Stemmed Words
332	11,264	10,849	8,371

The Training/Testing Split: To produce the model, the data sets (positive and negative) had to be separated into two parts: training set and the testing set. The tool allows the user to control the split either by ratio or by number of examples. Although we tried several splits but we settled with the tool's default ratio which is 70/30 which resulted in generally better classification effectiveness measured by accuracy, precision, recall, and F-Measure.

The Text Classification (TC) Process: The classical Text Classification approach has several main steps namely: Feature Extraction (FE), Since the sports news have been structured into folders, the data has been ready for classification with the following stages: Reading the files, Tokenization (which is equivalent to Feature Extraction), Stop Word removal, stemming, further word filtering and vectorization. Feature Selection (FS): Chi-Square was used. The Model Generation stage consists of two parts training and testing. The training part consists of the classification algorithm which generates the required model while the testing part consists of two sub-processes: applying the model and measuring its performance. The model is generated in binary format for further use on different data sets.

3. Constructing the Models

Three ML-based techniques were used for this experiment to classify cricket sports news. The following is the results of each one. Note that each experiment was run three times and the results were averaged.

SVM: This SVM is based on the evolutionary implementation. With default settings and dot (linear) kernel selected. The following table shows results of the four cases:

Table 3. SVM Experiment Results

Combination	Precision %	Recall %	F-Measure%	Accuracy %	Model Size	Dimensions	Time
Plain	100	97.92	98.95	99	181.6	11264	11:38
+Stop Words	100	100	100	100	170.9	10849	10:05
+Stemming	100	97.92	98.95	99	135.9	8371	08:15
+Chi-Square	100	100	100	100	18.5	837	00:50

Observations: Precision is not affected by any feature selection technique. Recall does respond positively but not to stemming. The only two factors that benefit from feature selection are time and model size. They are directly proportional to the number of features. The number of support vectors is not sensitive to the number of features (dimensions). They are as follows: 315, 309, 315, and 309.

C4.5: Default settings were employed such as pruning with confidence threshold of 0.25. The following is the results of the four cases:

Table 4. C4.5 Experiment Results

Combination	Precision %	Recall %	F-Measure%	Accuracy %	Model Size	Dimensions	Time
Plain	95.92	97.92	96.91	97	48.2	11264	22
+Stop Words	95.92	97.92	96.91	97	46.5	10849	21
+Stemming	100	97.92	98.95	99	33.9	8371	16
+Chi-Square	100	97.92	98.95	99	3.6	837	14

Observations: Stemming is the only feature selection technique that improves precision and f-measure while recall remains constant at 97.92. Timings slightly improve while model size is dramatically affected by feature reduction. Tree sizes are respectively: 19, 21, 11, and 11 while numbers of leaves respectively are: 10, 11, 6, and 6.

Naive-Bayesian: The following table shows the results of the four cases:

Table 5. Naive-Bayesian Experiment Results

Combination	Precision %	Recall %	F-Measure%	Accuracy %	Model Size	Dimensions	Time
Plain	96	100	97.96	98	36.8	11264	15
+Stop Words	97.96	100	98.97	99	35.5	10849	18
+Stemming	97.96	100	98.97	99	27.4	8371	12
+Chi-Square	97.96	100	98.97	99	2.9	837	11

Observations: The table shows that the Naive-Bayesian method effectiveness is almost not affected by feature selection. Precision shows a 1% improvement using stop word removal. Only the size of the model becomes smaller and is directly proportional to the number of dimensions. Timings show slight improvement with feature selection.

4. Analysis

Employing the F-Measure effectiveness measure, the following table is constructed for all three ML methods used in the experiment:

Table 6. F-Measure Effectiveness

Combination	SVM	C4.5	Naive-Bayesian
Plain	**98.95**	96.91	97.96
Stop Word Removal	**100**	96.91	98.97
+ Stemming	98.95	98.95	**98.97**
+ Chi-Square FS	**100**	98.95	98.97

In general, all methods are top performers and are capable of doing the job. Results range between 95.74% and 100%. This implies that none of them could be discarded as being weak or for showing dissatisfactory effectiveness results only (depending on the application).

Further we need to look at each feature selection combination. Recall we have three cases in addition to the default (plain) case which we would start with. Table 6 highlights the top F-Measure value in each combination in bold.

- In the plain case: No feature selection technique is employed and all features (terms) are used (No. of terms = 11,264). The order of the methods from top to bottom according to their F-Measure values is: SVM, Naïve-Bayesian, and C4.5.

- The second case: When stop word removal is being applied, the number of features (terms) is reduced almost 10% down to 10849. The order of the methods according to their F-Measure values has not changed from the plain case but some values did. All increased but the C4.5 remained as is.

- The third case was to combine both stop word removal and stemming. So as a result, the number of features has dropped to 8371 from 11264 (almost a 25% drop). The order of methods according to the F-Measure values is as follows: Naïve-Bayesian, then SVM, C4.5 Tie. It is important to note that SVM scores less in this case which supports the claim that stemming has negative or no impact on effectiveness. On the other hand, C4.5 shows improvement which support counter arguments that stemming have positive impact on effectiveness.

- The fourth and last case where stop word removal, stemming, and only top 10% features are selected using the chi-square statistical method (measures the lack of independence) the number of features is down to 837 almost 7.5% of the total features. The result of the experiment shows that the final order of the methods according to their F-Measure values is as follows: SVM, Naïve-Bayesian, and C4.5. SVM seem to be more sensitive to feature selection based on Chi-Square method.

It was observed that the C4.5 method did not excel in any of the combinations in terms of F-Measure in our experiment.

Such high values of f-measure (100%) sometimes do come at an expense in both execution (learning) time and resultant model size. Observe the following table:

Table 7. Performance (Times)

Combination	SVM	C4.5	Naive-Bayesian
Plain	11:38	00:22	**00:15**
+Stop Word Removal	10:05	00:21	**00:18**
+Stemming	08:15	00:16	**00:12**
+Chi-square	00:50	00:14	**00:11**

In General, performance has a totally different view than effectiveness when it comes to the differences. There is a wide variance between the lowest and the highest performer: 11 seconds in the case of Naïve-Bayesian compared with 11 minutes and 38 seconds for SVM which is almost 70 times. Note that there are very well known superfast SVM implementations such as SVMLight, LibSVM, and mySVM which are not part of this experiment but covered in other papers which we will compare with later[8].

Based on the discussion on F-Measure, we combine it with the timing to determine the best effectiveness/performance ratio for each case:

- In the plain case, the order of the methods according to the recorded times is: Naïve-Bayesian, C4.5 which are all sub-minute then comes SVM at above 10 minutes. There is a clear gap between the three methods.

- The Stop word removal case: all numbers improve almost equally and the order of methods is the same as in the plain case.

- The third case: Stop word removal and stemming: Analyzing effectiveness showed mixed results, efficiency has improved for all methods by at least 20% (proportional to the number of features removed). The order of methods in this case is: Naïve-Bayesian, C4.5, and SVM.

- The fourth case: Stop word removal + Stemming + 10% Chi-Square Feature Selection: The difference in

time among the three methods has narrowed dramatically from order of magnitudes to almost 4 to 5 times. The order of methods is: Naïve-Bayesian, C4.5, and SVM. Efficiency improved slightly for both Naïve-Bayesian, C4.5, while improved 10 times for SVM (proportional to the number of features removed). This also supports the finding from effectiveness analysis that SVM family responds positively to feature selection based on Chi-Square techniques. Timings are depicted below in figure 1.

By a quick look at the size of the resultant models, all model sizes responded to the fourth case and are almost one tenth of their size. It was observed that only Naïve-Bayesian excelled in terms of efficiency measures in our experiment. As a conclusion, Naïve-Bayesian seems to posses the right balance of both efficiency and effectiveness followed by C4.5. It is hard to recommend SVM without proper feature selection techniques.

Comparing those results with an earlier study on a different set of ML based classifiers [8], LibSVM showed the best efficiency and effectiveness results running on the same corpus. By comparing it with Naive-Bayesian classifier we get the following results:

Table 8. Comparing Top Classifiers Performance

Combination	LibSVM	Naive-Bayesian	LibSVM	Naive-Bayesian
Plain	**98.95**	97.96	00:21	**00:15**
+Stop Word Removal	98.95	**98.97**	**00:12**	00:18
+Stemming	98.95	**98.97**	**00:09**	00:12
+Chi-square	**100**	98.97	00:12	**00:11**

Numbers are very close. So we need to use statistical significance to see if there is any real difference between them. Both t-test and ANOVA (required by the tool) were used based on the f-measure and 95% confidence level and showed in all four cases that there was no significant difference between the two methods.

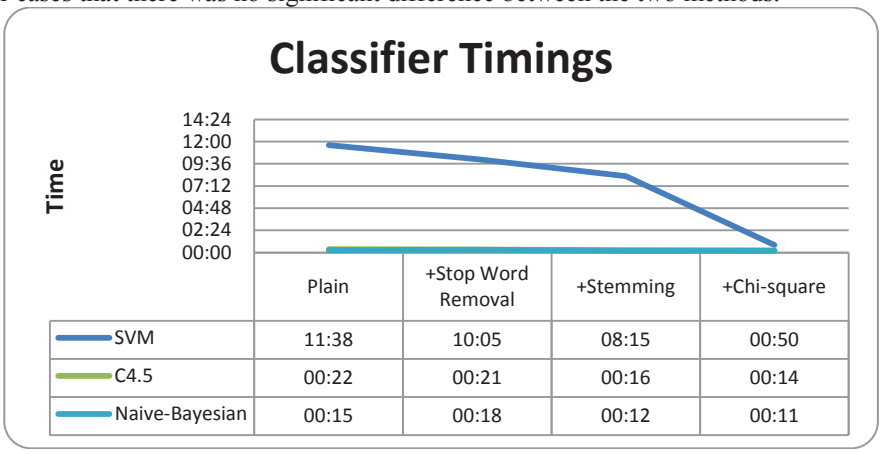

Fig. 1. Timing Comparisons.

5. CONCLUSION

Three binary text classifiers were built to test the cricket class of SGSC. Their effectiveness was measured using four chosen measures namely, Precision, Recall, Accuracy, and F-Measure. Additional measures such as time and model size were discussed to find the most suitable algorithm. Three variations of feature selection cases were performed along with a plain case. Naïve-Bayesian leads the pack with best effectiveness ratios overall. This experiment also demonstrates that such experiments are possible using COTS and open-source SW running on mainstream HW to conduct what used to be a specialized controlled only experiment.

ACKNOWLEDGMENT

The authors wish to thank the College of Computer and Information Sciences and the research center in the College of Computer and Information Sciences, King Saud University for their partial funds of this work. We also thank Eng. Mohammad Amin for his support and assistance in RapidMiner.

References

1. S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, N. Khorsheed, and A. Al-Rajeh. Automatic Arabic Text Classification. JADT 2008.
2. D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research 5 (2004): 361 – 397.
3. T. Rose, M. Stevenson, and M. Whitehead. The Reuters Corpus Volume 1 – from Yesterday's News to Tomorrow's Language Resources. Reuters.
4. Y. Yang and J. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *International Conference on Machine Learning*, pages 412–420, 1997.
5. T. Zakzouk and H. Mathkour. Building A Corpus for Text Classification. *The 2010 International Conference on Intelligent Network and Computing* (ICINC 2010).
6. F. Colas and P Brazdil. Comparison of SVM and some older classification algorithms in text classification tasks, *In Proc. of the Conference on Artificial Intelligence in Theory and Practice, International Federation for Information Processing,* 217:169-178, 2006.
7. Y. Yang and X. Liu. A re-examination or text categorization methods. In *proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information retrieval*, Pages 42-49, 1999.
8. T. Zakzouk and H. Mathkour. Text Classifiers for Cricket Sports News. In *proceedings of International Conference on Telecommunications Technology and Applications ICTTA 2011*, Pages: 196 - 201. Sydney, Australia.